Exhibit 4

Copy of an Internal, Confidential, and Non-Publicly Disclosed Patent Predisclosure Document Written by Inventor Ikko Fushiki, entitled, "137879.1 – XsRGB Standard and Implementation"

(Dates Redacted)
(5 pages)

137879.1 – XsRGB Standard and Implementation – Patent Disclosure

Ikko Fushiki

Prior Disclosure

May 5, 1999, Meeting with HP: We exchanged the idea of xsRGB and proposed the collaboration.

May 12, 1999, Meeting with Corbis: We illustrated the outline of xsRGB scheme and suggested Corbis adapt xsRGB for their image format.

Introduction

Since the introduction of color devices for personal computers such as color monitors, color scanners, digital camera, and color printers, it became important for computer systems or applications to be able to produce consistent colors among different devices. In response to that demand, Microsoft and Apple introduced ICM and ColorSync, respectively. Non-OS venders like Kodak have their own color management systems. Each color management system has its own color reference frame.

Here we have invented the scheme to extend the existing sRGB to the easier format, to the higher accuracy, and to the wider range of color space (gamut). At the same time we are extending the RGB format so that it fits to the advanced graphics system with alpha channel blending capabilities. The co-inventors of this patent are Ikko Fushiki, Hock San Lee, and J. Andrew Gossen.

Motivation for Invention

Most color management systems including Microsoft and Apple are based on RGB-space with certain types of monitor properties. Microsoft adapted sRGB as the standard color reference frame. This works fairly well among monitors. However, there are criticism of this format by color device venders and publishing industry in the narrow ranges of the gamut (the displayable range of the color space) and in the low accurary of sRGB data. The gamut of sRGB does not cover the displayable colors of all monitors, printers, nor scanners. The storing the image data in 8 bit per color channel is not good enough when the calibration or image processing is applied to the image. Indeed the recent scanner or digital cameras can take images in more than 8-bit in each color channel. We need to have a standard to be able to handle the data in higher accurary.

The advanced graphics system requires the anti-aliases and blending effects. Those effects are handled by an extra component called "alpha channel". In order to perform the anti-alias and blending operation correctly with the alpha channel, we need to have the linearized color component in terms of their intensities. However, sRGB or other

color management referece color space store the color values in non-linear way. It's non-linearity is expressed usually as "gamma" value. The gamma values of Microsoft and Apple's color management systems are 2.2 and 1.8, respectively. This was important to keep the data in non-linear way when the each component is restricted to 8 bit. When the non-linear 8 bit data is converted to the 8 bit linear value, it tends to create a large gap in the lower intensity values and the resulting images tend to show contours. However, when the size of each component is extended to higher bit (12 bit or higher), we no longer have this restriction. We can certainly store the data linearly in 16 bit component without having noticeable contour effects.

Problem Addressed

We have addressed the following problems:

1. sRGB has narrow gamut.
2. The precision of sRGB is not accurate enough when the color calibration or image processing is applied to the image.
3. The blending and anti-alias operation needs the linear RGB.
4. The current color management is complicated.
5. We need a standard format for the image archive with consistent colors without degrading the original data.

We have solutions for the above problem. We call our new format XsRGB.

1. XsRGB allows the component of each color to be negative and to be beyond 1 (when normalized to 1 in sRGB). XsRGB's gamut is larger than visible color space. Hence, XsRGB can express any color of any device.
2. XsRGB stores each component in 16 bit or higher (possibly in floating point for 32 bit).
3. XsRGB's each component is linear (gamma = 1).
4. The color profile venders do not have to clip to the narrower gamut and they do not have to non-linearize to create XsRGB. This avoids the confusion of different gamma values in different color standards.
5. The standard images store XsRGB values without attaching ICC profile.

Description of the Invention

XsRGB's definition starts with the relation with linearized sRGB and 1931 ICE XYZ values. Let R_0, G_0, and B_0 denote the normalized red, green, and blue components, respectively. A sRGB device can produce values between 0 and 1. Let X, Y, and Z denote 1931 ICE XYZ values but they are normalized to 1. The relationship between the normalized XsRGB and XYZ are given by

$$\begin{pmatrix} R_0 \\ G_0 \\ B_0 \end{pmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \qquad (1)$$

When we allow each component to go from – 4 to 4, this covers more than the whole gamut. The proof of this argument is discussed in the document "Extension of sRGB color and GDI+". We can use the definition (1) as the floating version of XsRGB. When we use 16 bit version of XsRGB, we must use at least 1 bit for a sign and 2 bits for the integer part. Hence, we use the remaining 13 bits for the decimal points. The 16 bit definition of RGB components of XsRGB is given by

$$\begin{pmatrix} R_{16} \\ G_{16} \\ B_{16} \end{pmatrix} = 8192 \times \begin{pmatrix} R_0 \\ G_0 \\ B_0 \end{pmatrix} \qquad (2)$$

Equation (2) is much simpler than the definition of 8 bit sRGB since no gamma corrections are involved.

The conversion from 16 bit XsRGB to 8 bit sRGB are as follows. Let C_{16} and C_8 denote one of components in 16 bit XsRGB and 8 bit sRGB, respectively. Their relationships are

$$\begin{array}{ll} C_0 \equiv C_{16}/8192 & \text{(This corresponds to the normalized linear XsRGB)} \\ C_8 = 0 & \text{for } C_{16} < 0 \\ C_8 = 12.92 \times C_0 & \text{for } 0 \le C_0 < 0.04045 \ (0 \le C_{16} \le 331) \\ C_8 = 1.055 \times C_0^{(1.0/2.4)} - 0.055 & \text{for } 0.04045 \le C_0 < 1 \ (332 \le C_{16} < 8192) \\ C_8 = 255 & \text{for } C_0 \ge 1 \ (C_{16} \ge 8192) \end{array} \qquad (3)$$

We have clipped blow 0 and above 8192 of 16 bit XsRGB when converting to 8 bit sRGB. This is our default and the easiest way. The color device venders can modify the clipping routines to produce the optimal 8 bit sRGB.

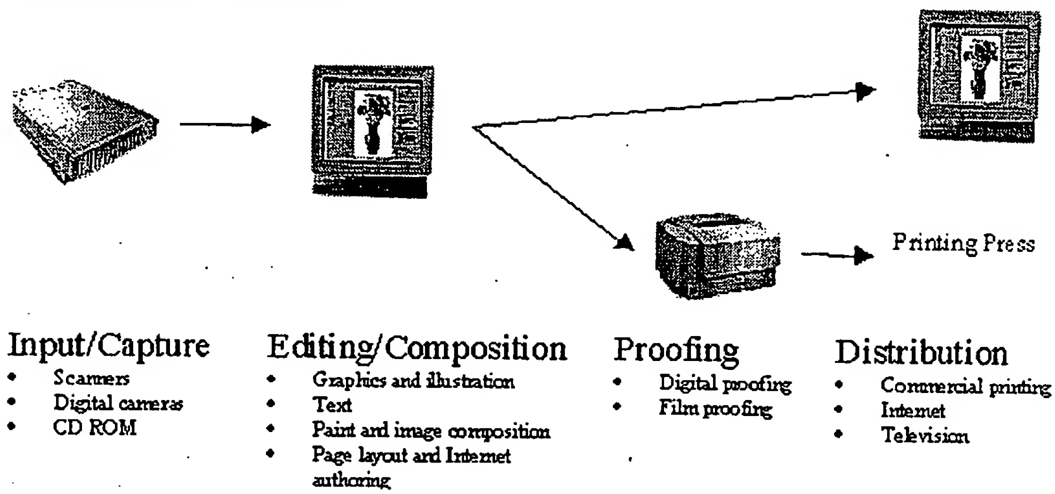
The reverse relation ships are:

$$\begin{array}{ll} C_{16} = 2.4865 C_8 & \text{for } 0 \le C_8 \le 10 \\ C_{16} = 8192 \times [(C_8 + 14.025)/269.025]^{2.4} & \text{for } 11 \le C_8 \le 255 \end{array} \qquad (4)$$

Our etension of sRGB has the following advantages. The blending operations with alpha channel can be directly applied to XsRGB since XsRGB is linear. The color device venders can easily create XsRGB profile from their CIE XYZ profiles. When XsRGB is

used for color reference, there is no need to rotate color components to display in 8 bit (each component) sRGB device. Only gamma correction described in Eq. (4) is necessary to convert to 8 bit sRGB. Even without the exact calibration, XsRGB will give fairly descent output for monitors. The scanned images can be stored in XsRGB format without loosing the bit depths since most scanners produce data in not more than 12 bit in each color component.

Diagrams and Flow Charts



This diagram is from http://www.microsoft.com/windows/platform/icmwp.htm.

The color device can have ICC profiles. Microsoft ICM reads those profiles and it converts colors when it sends to other devices when a user chooses to do color calibration. We would like to extend the ICC profiles so that the color devices can have XsRGB profiles. This gives the higher accuracy and unlimited color gamut. All the images are stored in XsRGB format. When the image is sent to the other device, the XsRGB values are converted to the values in the device color space or the color device can handle XsRGB values by itself.

Once images are obtained in the computer from input devices like scanners and digital cameras or from CD ROM and Web, the images are converted to XsRGB format to do blending operations. Then the synthesized images are sent to the output devices. During those process, we can make XsRGB as the standard color space.

Related Writings and Products

Fushiki, I. 1999, "Extension of sRGB color and GDI+", attatched.

IEC. 1998, "Default RGB color space – sRGB", http://www.srgb.com/sRGBstandard.pdf

Microsoft, "Color Management in Microsoft Windows Operating Systems", http://www.microsoft.com/windows/platform/icmwp.htm

Starkweather, G., "Colorspace Interchange Using sRGB", http://www.srgb.com/sRGBColorSpacePaper.pdf

Foley, J. D., et al. 1990, "Computer Graphics: Principles and Practice" (2nd ed.; Addison-Wesley), Chapter 13.

Apple Computer, Inc. 1995, "Advanced Color Imaging on the Mac OS" (1st ed.; Addison-Wesley).

Microsoft Product

This XsRGB will be used in GDI+. GDI+ may be used in the next version of Office and Vforms.

Record of Invention

March 2, 1999: ikkoF wrote a document of "Extension of sRGB color and GDI+" and discussed this idea in GDI+ meeting.

March 22, 1999: ikkoF disclosed the idea of XsRGB to HP (Mary Nielson et al).

May 4, 1999: ikkoF and andrewGo agree with the format of XsRGB.

May 5, 1999: ikkoF disclosed the idea of XsRGB to HP (Michael Stokes et al) and discussed each version of XsRGB format.

May 7, 1999: ikkoF and Michael Stokes agree with the importance of using the linear XsRGB. darrynD and sriramS discussed the marketing strategy.

May 12, 1999: ikkoF and hockL disclosed the idea of XsRGB to Corbis in the marketing and technical meeting. darrynD and sriramS discussed the corporation strategy with Corbis.

May 18, 1999: Held the disclosure meeting.